

The Delphion  
Integrated  
View

Other Views:  
INPADOC

Title: **JP5066984A2: RECOVERY METHOD FOR INDEX IN DATA BASE**

Country: **JP Japan**

Kind: **A**

Inventor(s): **NAGAE HIROAKI  
KANAI MISA**

Applicant/Assignee

**Inquire Regarding  
Licensing**

**FUJITSU LTD**

News, Profiles, Stocks and More about this company

Issued/File Dates:

**March 19, 1993 / Sept. 10, 1991**

Application Number:

**JP1991000229263**

IPC Class:

**G06F 12/00;**

Abstract:

**Purpose:** To execute the recovery of indexes at high speed.

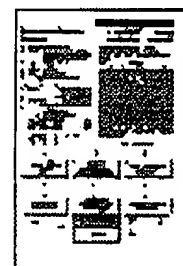
**Constitution:** Corresponding to the capacity of indexes, update history (log) areas are quantitatively allocated. Then, the amount of updating the indexes is measured and when the updating amount is more than the capacity of the allocated log areas, it is stopped to acquire the log of the index. Then, the recovery is executed by re-preparation and when there is margin in the log area, the recovery of the indexes is executed by using the log saved up to the moment. A shadow page (work area) is provided for updating the indexes, the indexes are updated in the shadow page, and only the updated result (before or after update) in the shadow page is defined as the log.

**COPYRIGHT: (C)1993,JPO&Japio**

Family: Show known family members

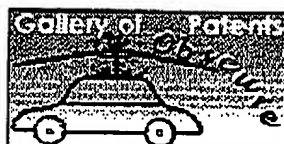
Other Abstract Info: none

Foreign References: No patents reference this one



View  
Image

1 p ge



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-66984

(43) 公開日 平成5年(1993)3月19日

(51) Int.Cl.<sup>5</sup>  
G 0 6 F 12/00

識別記号 庁内整理番号  
5 3 1 R 8944-5B

F I

技術表示箇所

審査請求 未請求 請求項の数3(全9頁)

(21) 出願番号 特願平3-229263

(22) 出願日 平成3年(1991)9月10日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 長江 宏晃

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72) 発明者 金井 美砂

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 井桁 貞一

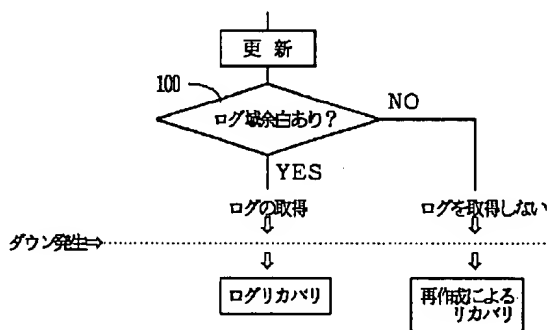
(54) 【発明の名称】 データベースにおけるインデックスのリカバリ方法

(57) 【要約】

【目的】 本発明は、データベースにおけるインデックスのリカバリ方法に関し、高速に、インデックスのリカバリを行う。

【構成】 インデックスの容量に対応して、一定容量の更新履歴(ログ)域を割当て、インデックスの更新量を計測して、該割当てたログ域の容量以上に、更新量が多くなった場合は、インデックスの更新履歴(ログ)の取得を止め、再作成によるリカバリを行い、ログ域に余白がある場合には、その時点までに待避しておいた更新履歴(ログ)を用いてインデックスのリカバリを行う。又、インデックスの更新に対して、シャドウページ(ワーク領域)を設け、該シャドウページでインデックスの更新を行うようにして、上記更新履歴(ログ)は該シャドウページでの更新の結果(更新の前か後か)のみとするように構成する。

本発明の一実施例を示した図(その1)



1

## 【特許請求の範囲】

【請求項1】障害発生時のデータベース(1)のインデックス(11)をリカバリする方法であって、

インデックス容量に対応して、一定容量の更新履歴(ログ)域を設定し、データ処理途上でのインデックス(11)の更新量を計測(100)して、該割当てた更新履歴(ログ)域の容量以上に、更新量が多くなった場合には、更新履歴(ログ)の取得を中止して、再作成によるインデックスのリカバリを行い、該更新履歴(ログ)域に余白がある場合には、その時点までに待避しておいた更新履歴(ログ)を用いてインデックスのリカバリを行うことを特徴とするデータベースにおけるインデックスのリカバリ方法。

【請求項2】上記更新履歴(ログ)を用いてインデックス(11)のリカバリを行う方法であって、該インデックス(11)の更新に対して、シャドウページ(ワーク領域)(①)を設け、該シャドウページ(①)上でインデックス(11)の更新を行うようにして、上記更新履歴(ログ)は該シャドウページ(①)での更新の結果(更新の前か後か)のみとすることを特徴とする請求項1に記載のデータベースにおけるインデックスのリカバリ方法。

【請求項3】上記インデックス(11)の更新履歴(ログ)をファイル記憶装置(2)に待避する際に、インデックス(11)の更新履歴(ログ)をバッファメモリ上に書き出して、システムの障害時等において、一括して、該バッファメモリから上記ファイル記憶装置(2)に書込むことを特徴とする請求項1に記載のデータベースにおけるインデックスのリカバリ方法。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、データベースのインデックスのリカバリ方式に関する。近年データベースに対し、使用者の増加、及びデータ量の増加に伴い、システムダウン等の障害発生後の業務再立上げ時間短縮のために、データベースのリカバリの高速化が要求されている。

【0002】該データベースのリカバリには、利用者データのリカバリと、インデックスのリカバリがあるが、該インデックスのリカバリに関しても、高速化が要求される。

【0003】

【従来の技術】図3は、データベースの構成例を示した図であり、図4は、データベースのインデックスの構成例を示した図である。

【0004】通常、図3に示したデータベース1を構築する場合、その利用者データ10【図3(a)参照】は、各キー項目(例えば、従業員番号、名前、生年月日、等)に関して、必ずしも、キー順になっていない為、該利用者データの検索を高速に行うのに、インデックス11を作成する必要がある。

2

【0005】該インデックス11は、利用者データの各キー項目毎に、該利用者データをキー順に並べた時の利用者データ10上の位置を示すタプル識別子(タプルID)(TID)を用いて、図3(b1)、(b2)に示されている如くのインデックス11を構築する。

【0006】該インデックス11は、例えば、データベース管理ユーティリティの1つであるインデックス作成コマンドを用いて作成される。該インデックス11は、ページを単位として、例えば、図4に示したように、木構造のバランストリーインデックス(B-TREE-INDEX)を形成し、キー項目(本図の例では、図3のデータベースの「生年月日」)の内容によって、高速に該当のインデックス11を含むページを検索できるようになっている。

【0007】そして、データベース(具体的には、利用者データ10)1の変更があると、データベース管理システムにより、上記バランストリーインデックス(B-TREE-INDEX)が更新される。

【0008】この場合、単に、あるページに、上記タプル識別子(タプルID)(TID)が追加される場合とか、該ページが満杯になっていて、同じ階層に新たにページを追加したり、下位階層のページを形成する必要がある場合等があり、それぞれの更新内容に対応して、更新履歴(ログ)、即ち、インデックスの差分が、例えば、ファイル記憶装置(ディスク)2に格納される。

【0009】該更新履歴(ログ)としては、例えば、ページの追加を行った場合、新ページ(D)の割付け、旧ページ(C)から新ページ(D)へのポインタの設定、旧ページ(C)のインデックスの一部の新ページ(D)への移転、平均化、上位階層のページ(A)から新ページ(D)のポインタの設定、新ページ(D)へのインデックス11の追加等がある。

【0010】このような、インデックス11の更新を行っているときに、障害が発生した場合、利用者データ10のリカバリと共に、インデックス11のリカバリが行われる。

【0011】該インデックス11のリカバリは、上記データベース管理ユーティリティの1つである、上記作成コマンド、リカバリコマンドを用いて行われる。

(1)再作成：利用者データのリカバリ後、その利用者データに基づき、上記インデックス作成コマンドで、インデックスを再作成する方式である。

【0012】(2)ログリカバリ：インデックスの更新履歴(以下、ログという)をログファイルに退避しておき、システムダウン等の障害時には、該ログによりインデックスの差分(変更部分)のみを修正することにより、インデックスを修復する方式である。

【0013】

【発明が解決しようとする課題】上記の従来の再作成／ログリカバリには以下の問題点がある。

3

(1) 再作成の場合：利用者データ量に応じて時間がかかる。特に、システムダウン時の更新量によらず再作成を行うため、利用者データ量が多い場合は遅い。

【0014】(2) ログリカバリの場合：前述のように、インデックス11のページ（入出力単位）、レコード操作毎にログが必要である。

又、該ログ管理（ログの作成、ログのディスクへの書き出し、ログと利用者データの同期化）の為に、プロセッサ(CPU)処理、及び、ディスクアクセスが共に増大し、アクセス性能が低下する。更に、更新量が多い場合にはログ量も増加し、インデックス修復時間が、上記再作成より遅くなることがある。

【0015】本発明は上記従来の欠点に鑑み、データベースシステムにおいて、障害が発生した場合の、業務再立上げ時間を短縮する為に、インデックスのリカバ리를高速化し、インデックス修復のための更新履歴取得によるアクセス性能の低下（CPU、ディスクアクセス）を防止することができるインデックスのリカバリ方法を提供することを目的とするものである。

【0016】

【課題を解決するための手段】図1、図2は本発明の一実施例を示した図である。上記の問題点は下記の如くに構成したデータベースにおけるインデックスのリカバリ方法によって解決される。

【0017】(1) 障害発生時のデータベース1のインデックス11をリカバリする方法であって、インデックス容量に対応して、一定容量の更新履歴（ログ）域を設定し、データ処理途中のインデックス11の更新量を計測100して、該割当てた更新履歴（ログ）域の容量以上に、更新量が多くなった場合には、更新履歴（ログ）の取得を中止して、再作成によるインデックス11のリカバリを行い、該更新履歴（ログ）域に余白がある場合には、その時点までに待避しておいた更新履歴（ログ）を用いてインデックスのリカバリを行うように構成する。

【0018】(2) 上記更新履歴（ログ）を用いてインデックス11のリカバリを行う方法であって、該インデックス11の更新に対して、シャドウページ（ワーク領域）①を設け、該シャドウページ①上でインデックス11の更新を行うようにして、上記更新履歴（ログ）は該シャドウページ①での更新の結果（更新の前か後か）のみとするように構成する。

【0019】(3) 上記インデックス11の更新履歴（ログ）をファイル記憶装置（ログファイル）2に待避する際に、インデックス11の更新結果をバッファメモリ上に書き出しておいて、システムの障害時等において、一括して、該バッファメモリから上記ファイル記憶装置（ログファイル）2に書き込むように構成する。

【0020】

【作用】本発明においては、下記の3つの方法で、インデックスのリカバリを行うようにする。

4

【0021】1) インデックスリカバリ方法の選択：従来は、システムダウン時のインデックスの更新量によらずインデックスを再作成していたが、本発明においては、インデックスの更新量を計数して、その更新量に応じて再作成するか、ログリカバリを行うかを選択するようにする。

【0022】具体的には、インデックス容量に応じて、一定のログ域を割当て、割当て以上にインデックスの更新量が多くなった場合は、その時点でログの取得を止め、再作成リカバリに切り換える方式をとる。

【0023】2) インデックスログの取得方法：インデックスを更新する場合、インデックスは分裂（ページの同階層、下位階層への追加）／縮退（ページの削除）を繰り返して、複数のページが影響を受けるため、リカバリログとしてそれ毎のログが必要となる。

【0024】そこで、本発明においては、インデックス更新途中の状態は仮メモリ（シャドウページ：ワーク領域）で管理し、更新終了後に更新前のページと入れ替える方法（シャドウページ方法）をとることにより、インデックスの更新前後のレコードだけでリカバリが可能である。そして、入替えは、ポインタの張替えによって行う。

【0025】3) ログとデータの入出力管理方法：インデックスレコード更新時、毎回ファイル記憶装置（ディスク）に書き出さず、システムダウン等の障害発生時に、それまでに、バッファ上に一時的に書き出されているデータ及びログデータをファイル記憶装置（ディスク）に書き出すように制御することにより、該インデックスリカバリに関連するファイル記憶装置（ディスク）への入出力回数を削減することができる。

【0026】上記のように、インデックスリカバリの方法を、更新データ量に応じて決定することにより、従来のように、更新量が少ないのに再作成によるリカバリを行ったり、更新量が多いのにログリカバリを行うということがなくなる。

【0027】又、ログリカバリにおいてシャドウページ（ワーク領域）を使用して、最低限インデックスリカバリに必要な情報のログ、具体的には、図2(b)に示したように、例えば、「更新種別（インサート／デリート）、インデックス（キー値、タプルID）」だけを取得することによって、ログ量を削減することができ、不必要な情報を作成、ディスクへの書き出しを行う必要がなくなる。又、インデックスレコード更新時、毎回ファイル記憶装置（ディスク）へ書き出すのを止めて、一時的にバッファ上に保持しておき、システムダウン時等に、一括して、該ファイル記憶装置（ディスク）に待避することにより、インデックスリカバリに関連する入出力回数が削減される。

【0028】

【実施例】以下本発明の実施例を図面によって詳述す

5

る。前述の図1、図2が本発明の一実施例を示した図であり、図1はインデックスリカバリ方法の選択方法を示し、図2はシャドウページを使用した更新前後のログだけを取得する方法を示している。

【0029】本発明においては、データベース1におけるインデックス11のリカバリ方法を選択するのに、インデックス11の容量に対応して、一定容量の更新履歴(ログ)域を割当て、インデックス11の更新量を計測して、該割当てたログ域の容量以上に、更新量が多くなった場合はログの取得を止め、再作成によるリカバリを行い、ログ域に余白がある場合には、その時点までに待避しておいた更新履歴(ログ)を用いてインデックスのリカバリを行う。又、インデックス11の更新に対して、シャドウページ(ワーク領域)①を設け、該シャドウページ①でインデックス11の更新を行うようにして、上記更新履歴(ログ)は該シャドウページ①での更新の結果(更新の前か後か)のみとする。又、インデックスレコード更新時、毎回ファイル記憶装置(ディスク)2へ書き出すのを止めて、一時的にバッファ上に保持しておき、システムダウン時等に、一括して、該バッファからファイル記憶装置(ディスク)2に待避する手段が、本発明を実施するのに必要な手段である。尚、全図を通して同じ符号は同じ対象物を示している。

【0030】以下、図1と図2とにより、本発明のデータベースにおけるインデックスのリカバリ方法を説明する。先ず、図1において、インデックス11のログ域を獲得するときに、最大元数を設定しておき、インデックス11を更新する毎に更新量、例えば、レコード数を計数して、上記、予め、設定してある最大元数と比較して、ログ域に余白があるか否かを判定する。(処理ステップ100参照)

通常、複数のアプリケーションプログラムが動作しており、例えば、瞬間的に、該更新量が、上記最大元数を超過したことが認識された場合には、それまで、保持していたログ情報を破棄して、以後のログの取得を中止し、障害発生時の前述の利用者データ10の更新データに基づいた、再作成方法でインデックスのリカバリを行う。

【0031】即ち、データベース管理ユーティリティの1つであるインデックス作成コマンドを発行して、インデックスのリカバリを行う。又、上記インデックス11の更新量が、上記最大元数より少ないことが認識された場合には、その時点までに取得されたログ情報に基づいて、これも、データベース管理ユーティリティの1つであるリカバリコマンドを発行して、インデックスのリカバリを行う。

【0032】次に、図2によって、本発明によるインデックスログの取得方法を説明する。図2において、(a1)(a2)はシャドウページによる更新方法を示し、(b)はログ情報の例を示している。

【0033】インデックス更新前の状態が、図2(a1)に

6

示した状態であったとする。即ち、前述のバランスドトリインデックスが上位のページAの下位の階層に、ページB、Cがあって、例えば、満杯のページCに、あるキー項目のインデックスを挿入(INSERT)しようとした場合、従来方式においては、「新規なページDの割付け、ページCからページDへのポインタの設定、満杯のページCのデータの一部を、上記新規に設定したページDに移し替えを行い、該新規なページDへ、上記挿入(INSERT)しようとしたインデックスを追加、更に、上位階層のページAからページDへのポインタの設定」といったログ情報を、前述のファイル記憶装置(ディスク)2に取得していたが、この従来方法では、ログ情報が膨大となり、そのログ情報のファイル記憶装置(ディスク)2への書き込み処理の為に、プロセッサ(CPU)処理、ディスクアクセスが共に増大し、アクセス性能を低下させてしまう問題があった。

【0034】そこで、本発明においては、図2(a2)に示してあるように、シャドウページ①をワーク領域として設けて、上記インデックス11の更新処理を、該シャドウページ①上で行い、該インデックス11の更新終了後、ポインタを張り替えることで、インデックスの更新前後のログ、例えば、前述の、「更新種別(インサート/デリート)、インデックス(キー値、タプルID)」

【図2(b)参照】だけを取得するように構成することで、ログ情報の大幅な削減を行うことができる。即ち、更新途中でログを取得しないようにして、図2(b)に示したフォーマットでのログのみを取得するようにする。

【0035】又、本発明によるインデックスリカバリ方法においては、利用者データ10、インデックス11のレコードを更新したとき、毎回、該更新データを、前述のファイル記憶装置(ディスク)2に書き出すことを行わず、例えば、主記憶装置(MSU)上のバッファ上に一時的に保持しておき、システムダウン時等において、該利用者データ10、及び、ログデータを、前述のファイル記憶装置(ディスク)2に書き出すように構成することで、該ファイル記憶装置(ディスク)2への入出力回数を削減することができる。

【0036】この場合、該障害が電源断等の場合には、該主記憶装置(MSU)上のバッファ上に保持されているログデータを保証する為の、例えば、バッテリー機構等が必要となる。

【0037】このように、本発明は、データベースにおけるインデックスのリカバリ方法を選択するのに、インデックスの容量に対応して、一定容量の更新履歴(ログ)域を割当て、インデックスの更新量を計測して、該割当てたログ域の容量以上に、更新量が多くなった場合はログの取得を止め、再作成によるリカバリを行い、ログ域に余白がある場合には、その時点までに待避しておいた更新履歴(ログ)を用いてインデックスのリカバリ

7

を行う。又、インデックスの更新に対して、シャドウページ（ワーク領域）を設け、該シャドウページでインデックスの更新を行うようにして、上記更新履歴（ログ）は該シャドウページでの更新の結果（更新の前か後か）のみとする。又、インデックスレコード更新時、毎回ファイル記憶装置（ディスク）へ書き出すのを止めて、一時的に、例えば、主記憶装置(MSU)上のバッファに保持しておき、システムダウン時等に、一括して、該バッファからファイル記憶装置（ディスク）に待避するようにした所に特徴がある。

【0038】

【発明の効果】以上、詳細に説明したように、本発明のデータベースにおけるインデックスのリカバリ方法は、インデックス容量に応じたログ域を設定し、ログ量を計数することにより、更新データ量に応じて、その状況に応じたインデックスのリカバリ方法、即ち、再作成／ログリカバリ方法を選択するようにする。又、インデックスの更新に、シャドウページ（ワーク領域）①を設けて、該シャドウページ①上でのみ更新作業を行い、ログ情報は、その結果のみ、即ち、挿入／削除したインデックスのみの情報とする。又、ログ情報のファイル記憶装置（ディスク）への待避は、システムダウン等の障害が発生したときのみに行うようにしたものである。従来は、システムダウンの発生時のインデックスの修復方法としてインデックスの再作成を行うか、ログリカバリを行うか選択方法があいまいであった為、更新量が多いのにログリカバリを行ったり、逆に少ないのに再作成をして非効率的なりカバリ処理をしてしまうことがあったが、上記のようにしてインデックスリカバリ方法を選択することによってその状況に適した方法でリカバリ処理を行うことができる。

8

【0039】又、ログリカバリを行う場合、従来のログの取得方法では不必要な情報まで取得されてしまっていたが、シャドウページを使用し更新前後の状態のログだけを取得することにより、ログ量が削減でき、ログ作成／ディスク書き出しのためのCPU処理、ディスク入出力回数を削減することができる。

【0040】更に、ログをバッファ上に待避しておき、システムダウン等障害発生時に、該バッファからファイル記憶装置（ディスク）に待避することにより、ファイル記憶装置（ディスク）に対する入出力回数を削減することができる効果が得られる。

【0041】従って、インデックスの修復時間が短縮し、システムダウン発生後の業務再立上げ時間を短縮させることができ、又、業務運用中のアクセス性能を高速化することができる効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例を示した図（その1）

【図2】本発明の一実施例を示した図（その2）

【図3】データベースの構成例を示した図

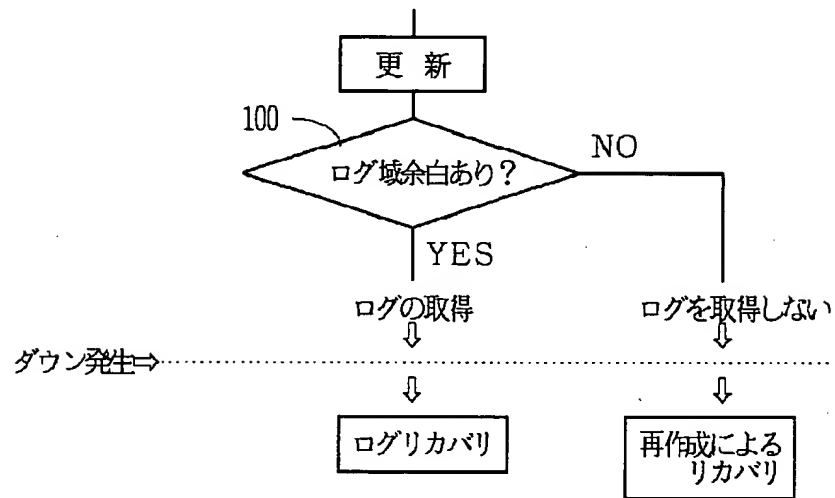
【図4】データベースのインデックスの構成例を示した図

【符号の説明】

1	データベース	10	利用者データ
11	インデックス		
100	処理ステップ		
2	ファイル記憶装置（ディスク）		
①	シャドウページ		
A, B, C, D, ~	バランスドトリーインデックスの各ページ		

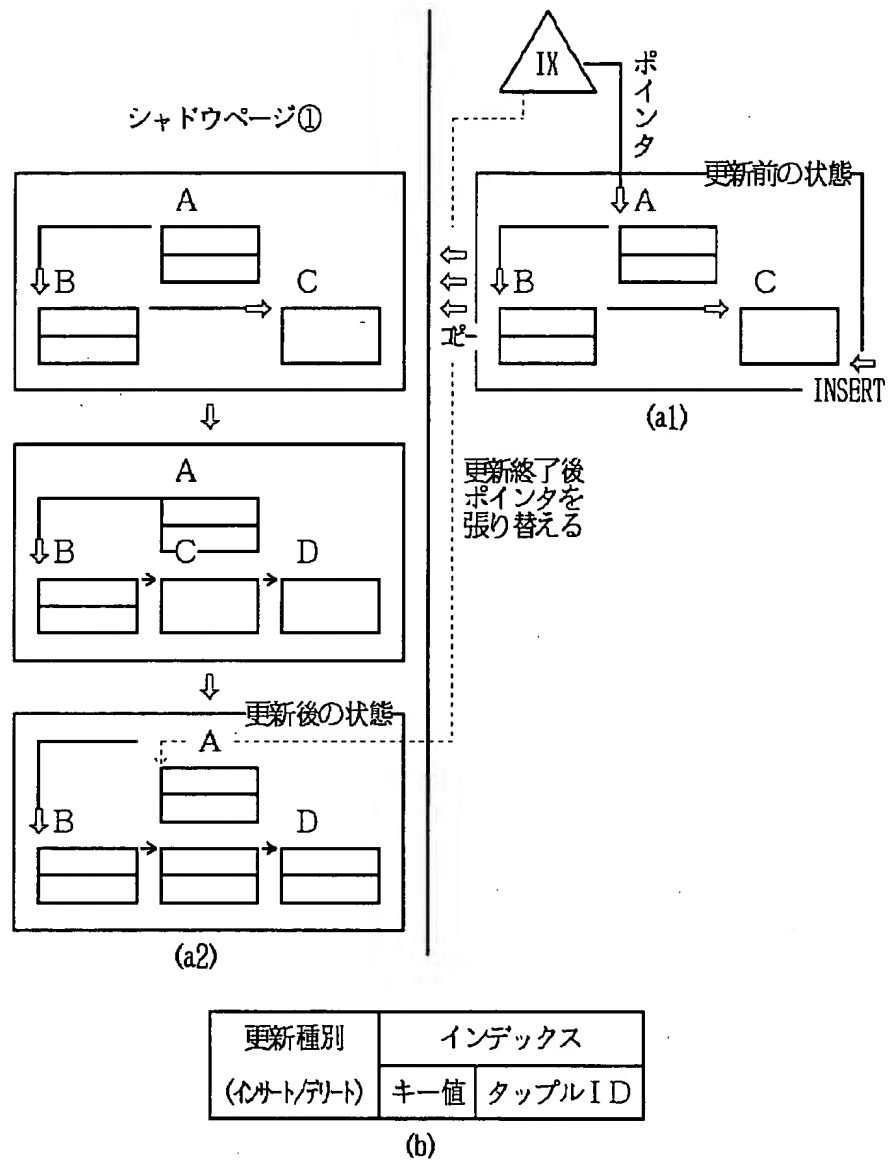
【図1】

本発明の一実施例を示した図 (その1)



【図2】

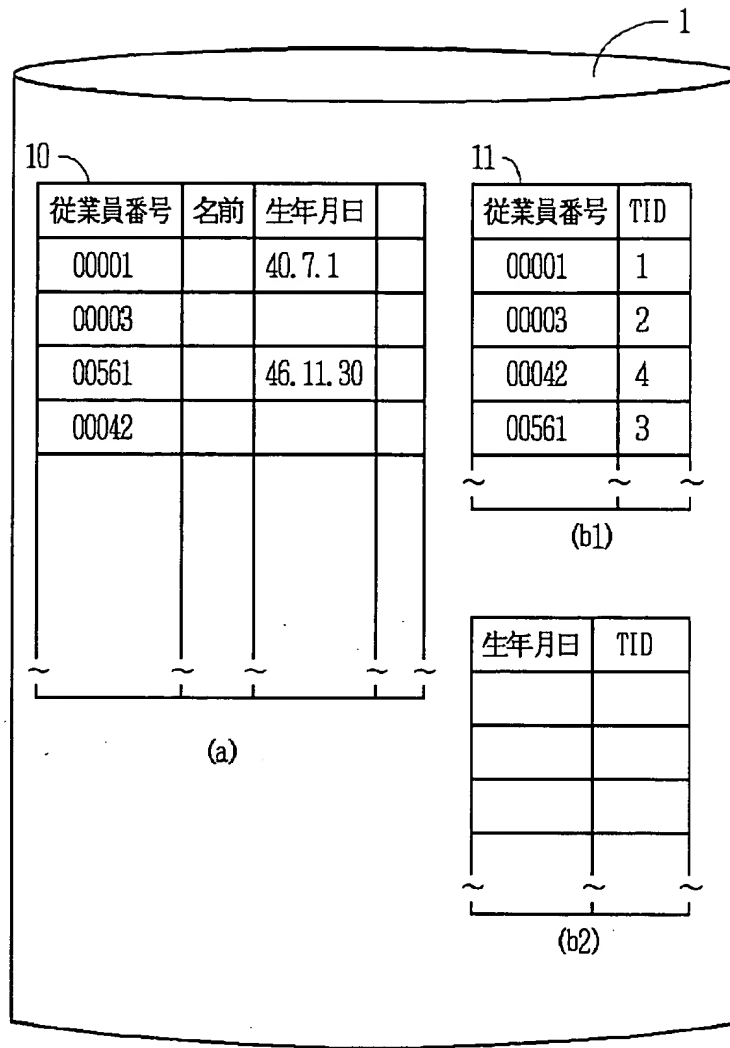
本発明の一実施例を示した図 (その2)





【図3】

データベースの構成例を示した図



【図4】

データベースのインデックスの構成例を示した図

